Modernization Partner

United States Department of Education
Student Financial Assistance

Application and Technical Architecture Standards
Executive Summary

Task Order #4
Deliverable #4.1.5

February 16, 2000

Executive Summary Overview

Department of Education

- ***Integration Application and Technical Architecture Standards***
  - **Introduction**
  - **Standards for Legacy Systems and Network Analysis Phase**
  - **Standards for Configuration Phase**
  - **Standards for the Build and Test Phase**

- *Data Warehouse Application and Technical Architecture Standards*
  - Introduction
  - Data Movement Standards
  - Data Warehousing Architecture Overview
  - Population Architecture
  - End-User Access

- *Internet Application and Technical Architecture Standards*
  - Introduction
  - Standards Covering Aspects of Development
  - Construction Standards

Introduction

The primary purpose of the Integration of these standards is to describe the model for the integration architecture application environment. The proposed standards will enable SFA to develop the Service Oriented Architecture recommended by Deliverable 4.1.2, Recommended Application Architecture.

Integration provides the technology services that enable the sharing of processes and data from disparate systems to support end-to-end business processes. The document proposes the standards with in the architecture to achieve integration between newly developed applications, such as the Internet and Data Warehouse, commercial off-the-shelf packages, and legacy systems.

The Integration Standards document applies to all systems within the SFA enterprise architecture that affect the achievement of the SFA Modernization Blueprint.

Standards for Legacy Systems and Network Analysis Phase

Department of Education

The level of analysis performed in this phase will depend on the level of understanding required for the downstream design activities to begin. For an integration architecture implementation using the IBM MQSeries Integrator application, the assessment will be key to the design, configuration, and build of the integration architecture.

It is also pertinent to realize the extent of the business process model produced during the assessment. This model is the central focus of the first phase of implementing a MQSeries Integrator system. The components of this model provide the design phase with the external systems that will connect to the MQSeries Integrator application, the direction of communication between them, and the types of messages each system will send or wish to receive.

Standards for Configuration Phase

Department of Education

The Configuration and Design Phase will be responsible for the design of the MQSeries Integrator product, the design of the technical architecture interfaces, the development and configuration of the required connections for the release, and the validation of the configuration. By building upon the previous work performed in the analysis phase, the design team formalizes and constructs the services and interactions to be performed by the automated business processes. Services and interactions are the basis of the MQ Series architecture and are supported by the three main components: MQSeries (messaging), MQSeries Integrator, and MQSeries Workflow

Standards for Configuration Phase (cont.)

Naming Standards:

Naming standards for directories, files, and variables will be applied to the design and development tasks involved in the MQSeries implementation.

Directory Names:

All servers and queues will have the same directory structures. This will ease the monitoring of multiple hosts and prevent system administrators from having the requirement to learn multiple installation setups to effective administer a host.

File Names:

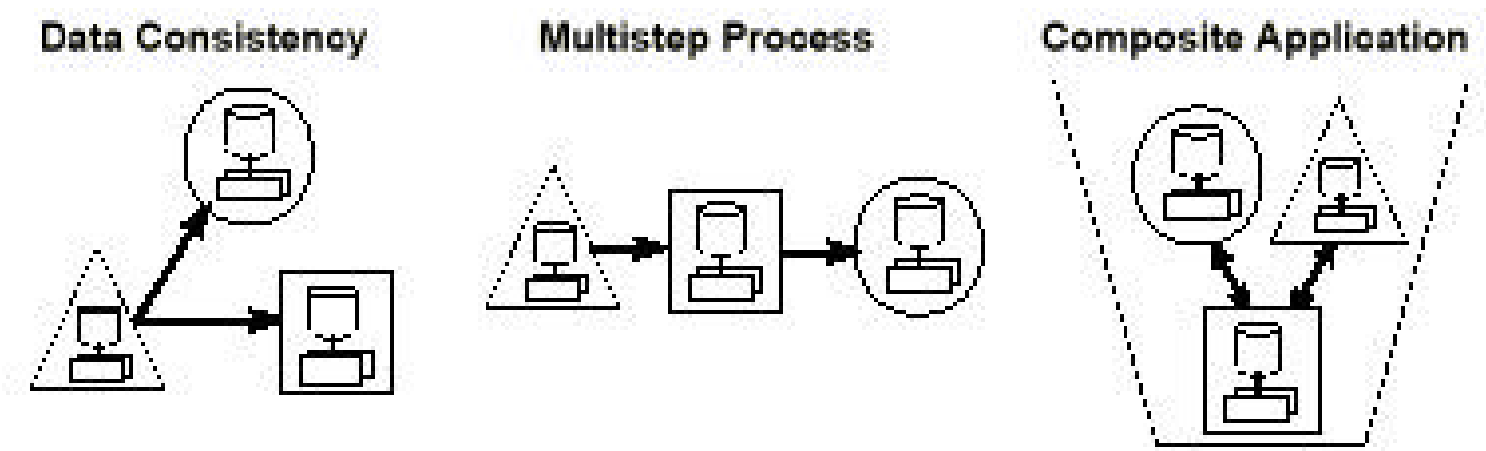
File naming standards will enable the effective use of variables and allow multiple analysts to understand the use of variables within the business flows of the system

Standards for Configuration Phase (cont.)

Department of Education

Integration Techniques:

The comprehension of the application integration techniques begin with the understanding of the three major types of application integration: data-centric, object-centric, and message-centric. Each style has its own strengths and limitations. The best solution for SFA's application integration will involve a combination of all three integration styles. SFA must understand the technique to achieve integration and determine which of the approaches to apply:



Standards for Configuration Phase (cont.)

Department of Education

Operational Data Store Connectivity:

For the Data Warehouse architecture all data will be received through MQSeries and written to the Oracle Operational Data Store (ODS) in Near-Real-Time to allow users access to the most current data. Therefore, the integration style required to support this requirement will be message centric (event-driven). The Oracle ODS once populated will perform an extraction, transformation, and load sequence periodically and directly populate the Oracle Data Warehouse and Data Mart in the form of batch files.

In order to provide access to the Oracle ODS, Data Warehouse and Data Mart, adapters will be used. These adapters are standard pre-built applications designed to handle communication to and from relational Oracle databases. It has the capability to extract metadata definitions of specified tables or views from the Oracle databases and generate a graphical tree representation of the column and row headings.

Standards for Configuration Phase (cont.)

Department of Education

Internet Connectivity:

The Client will make data requests to back end legacy databases or system applications using a Web browser front end. To ensure complex read and data updates are processed, the Composite Application Technique must be implemented. This technique binds applications together by business logic and methods over a program-to-program mechanism (CORBA ORB) to access the back-end data and applications.

For the internet architecture, all data requests will be received through MQSeries Integrator and passed on to the destination back-end systems. The requests will be processed in Near-Real-Time. The web browser users an EJB application server to process and forward a client's requests in the form of business events to the Integrator application.

Standards for Configuration Phase (cont.)

Department of Education

Legacy Connectivity:

SFA has a number of distributed legacy systems throughout their present environment, which will need to be integrated with MQSeries Integrator. To ensure these data and services are maintained in the future SFA architecture, Integrator will utilize the CICS adapter to communicate with the legacy applications, and ODBC adapters to exchange data directly with legacy relational databases

Intelligent Queuing Layer:

MQSeries queues act as powerful facilitators of reliable interprocess communication. The intelligence arises from the persistent recording of message state information necessary to ensure that subscribers acquire the messages they expect, in the proper sequence and without the risk of duplication - even when recovering from hardware failure. It exposes APIs which allow users to implement queues and interface with libraries written for various datastore implementations.

Standards for Build and Test Phase

The Build and Test Phase of the integration architecture will complete the technical architecture services for the implementation of the automated business applications. The Build and Test Phase of the integration architecture is concerned with the development, execution, and operations architectures.

Development Architecture:

The development Architecture provides the environment for the component-based solutions that supports the analysis, design, and construction phases of the MQSeries Integrator development process. The Integrator toolset provides GUIs for all major stages of configuration and design.

Execution Architecture:

The execution architecture is a unified collection of run-time technology services, control structures, and supporting infrastructure upon which the MQSeries Integrator application runs. The execution architecture components include MQSeries Queues, adapters, a core rules engine for data translation and transformation, and workflow components.

Operations Architecture:

The operations architecture is a combination of tools, support services, procedures, and controls required to keep the production system up and running efficiently. This environment provides network and systems performance monitoring, and diagnosis and failure reporting.

Executive Summary Overview

Department of Education

- *Integration Application and Technical Architecture Standards*
  - Introduction
  - Standards for Legacy Systems and Network Analysis Phase
  - Standards for Configuration Phase
  - Standards for the Build and Test Phase

- ***Data Warehouse Application and Technical Architecture Standards***
  - **Introduction**
  - **Data Movement Standards**
  - **Data Warehousing Architecture Overview**
  - **Population Architecture**
  - **End-User Access**

- *Internet Application and Technical Architecture Standards*
  - Introduction
  - Standards Covering Aspects of Development
  - Construction Standards

Data Warehouse - Introduction

Department of Education

The Data Warehouse Standards document defines both a standard framework, along with guidelines and standards for building data warehousing infrastructures at SFA. This document is intended to serve as a guide for future data warehousing development teams.

The Data Warehousing Standards and Guidelines cover the following areas:

- A recommendation on data warehousing standards
- Descriptions of key data warehousing terms and concepts
- An overview of the data warehouse end-to-end architecture

The document is organized around two primary areas:

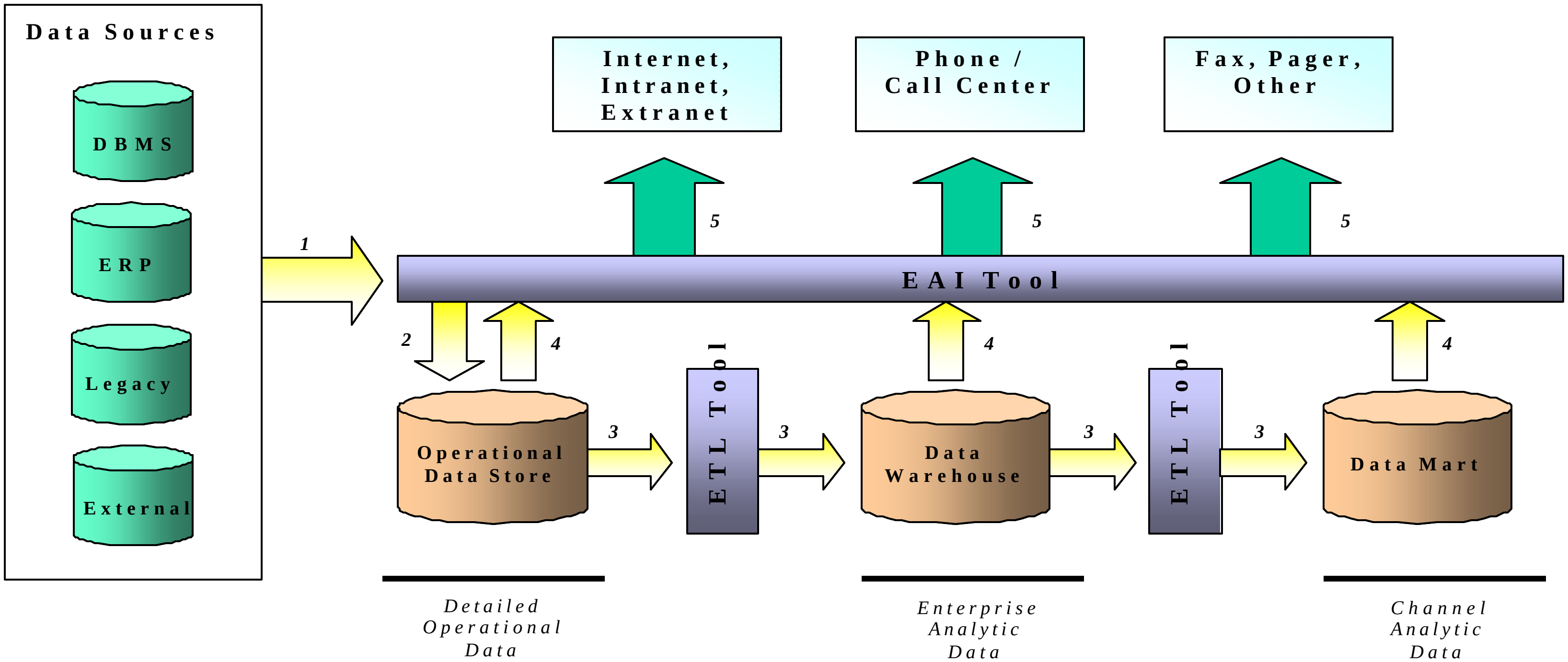
- Architectural standards and recommendations, based on Andersen Consulting's Business Integration Methodology, our industry experience and best practices, and our knowledge of SFA's objectives, initiatives, and environment
- Data warehousing terms, concepts and architectures

Data Movement Standards

Department of Education

In this section, recommendations on data movement standards and what tools to use have been made. Specifically, standards regarding how data is written to repositories, and how data is distributed and disseminated.

The figure below provides a snapshot view of data movement from source to target systems:



Data Movement Standards (cont.)

Department of Education

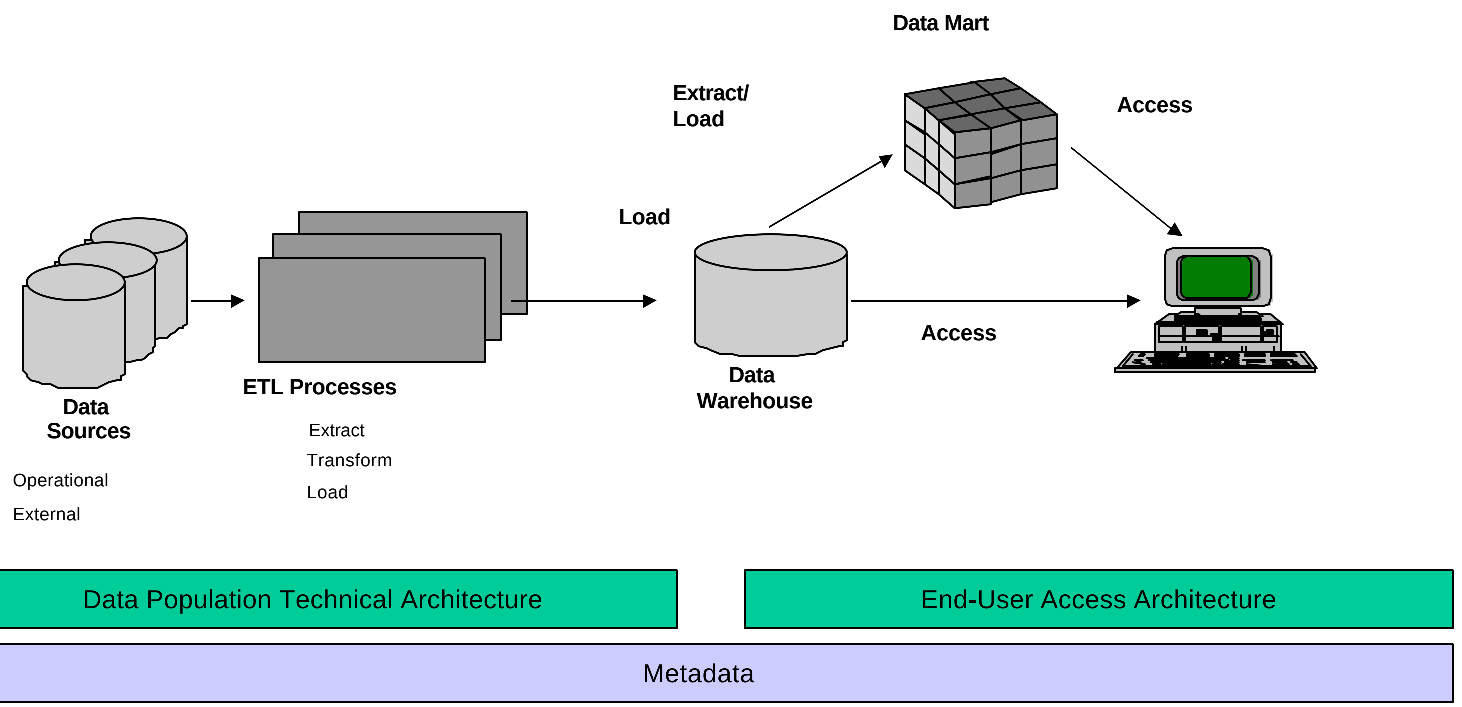
High level data movement standards include:

- Use the EAI architecture and tool to extract data from current transactional systems, and to load it into the Operational Data Store. EAI can provide near-real-time extracts and loads, as contrasted with batch extracts and loads with the traditional ETL tools. The "currency" of data will be important for certain reports, web and call center activities.
- Use ETL (Extract-Transform-Load) tool to extract data from the ODS and load it into the data warehouse. Also use ETL to "pump" data from the data warehouse to data marts. The ETL tool excels at transformations such as calculations and aggregations. ETL also provides robust metadata functionality, which gives users of the warehouse specific information about the data.
- Most users, no matter what the access channel, will gain access to the data warehousing repositories through the EAI (middleware) architecture. EAI will handle security and routing such that the right data gets to the right users from the right repository or systems. The exceptions are system, database, and data warehouse administrators, and some power users.
- Fall all access channels (phone, web, pager, PDA, etc.) wanting access to data warehouse repositories, EAI will access the information from the right repository and serve it up to the proper access channel.

Data Warehousing Architecture Overview

Department of Education

This section provides an overview of key terms and concepts within a data warehousing end-to-end architecture. It is structured within a conceptual model called the Data Warehouse Architecture Framework. This framework exists above the physical implementation level, although physical architecture choices and alternatives are discussed.



Data Warehousing Architecture Overview (cont.)

Department of Education

Business Metadata:

Business Metadata is the information needed by end-users to be confident in the meaning, quality and timeliness of the data. Without this information, the most technically robust and high-performing data warehouse will not be used to its fullest potential. A brief list of important business metadata is as follows:

- Business rules describing what is and is not included within the data warehouse
- Definitions of business hierarchies and key performance indicators (KPIs)
- Common business definitions and calculations for data elements
- Transformation and conversion rules in business context
- Source systems names/locations
- User security profiles

Metadata management consists of the processes that perform the definition, collection, control and publication of appropriate metadata to the right people at the right time. Determining what kind of metadata should be captured, how it should be published and what degree of integration is necessary is all part of this process.

Data Warehousing Architecture Overview (cont.)

Department of Education

Metadata Collection Strategies:

Metadata needs to be defined, stored and accessed in a comprehensive manner, just like the data in the warehouse. The approach to storage and access constitutes the metadata collection strategy.

A metadata collection strategy has the following attributes:

- *Integrated vs Non-Integrated*
- *Centralized vs Distributed*
- *Automated vs Manual*

Collection strategies typically consist of one or a combination of the following:

- *Metadata Repositories:* Utilized to integrate disparate metadata from multiple sources (tools, databases and processes) via automated feeds into a single point of collection
- *Decentralized Metadata Capture:* The use of an integrated metadata repository does not frequently occur for smaller-scale or tightly budgeted projects due to cost, complexity, and requirements not being strong enough to support this feature.
- *Manual Metadata Collection Techniques:* Even if centralized and automated methods of metadata collection cannot be utilized on a project, there are still high-benefit but manual-intensive methods of gathering and publishing metadata.

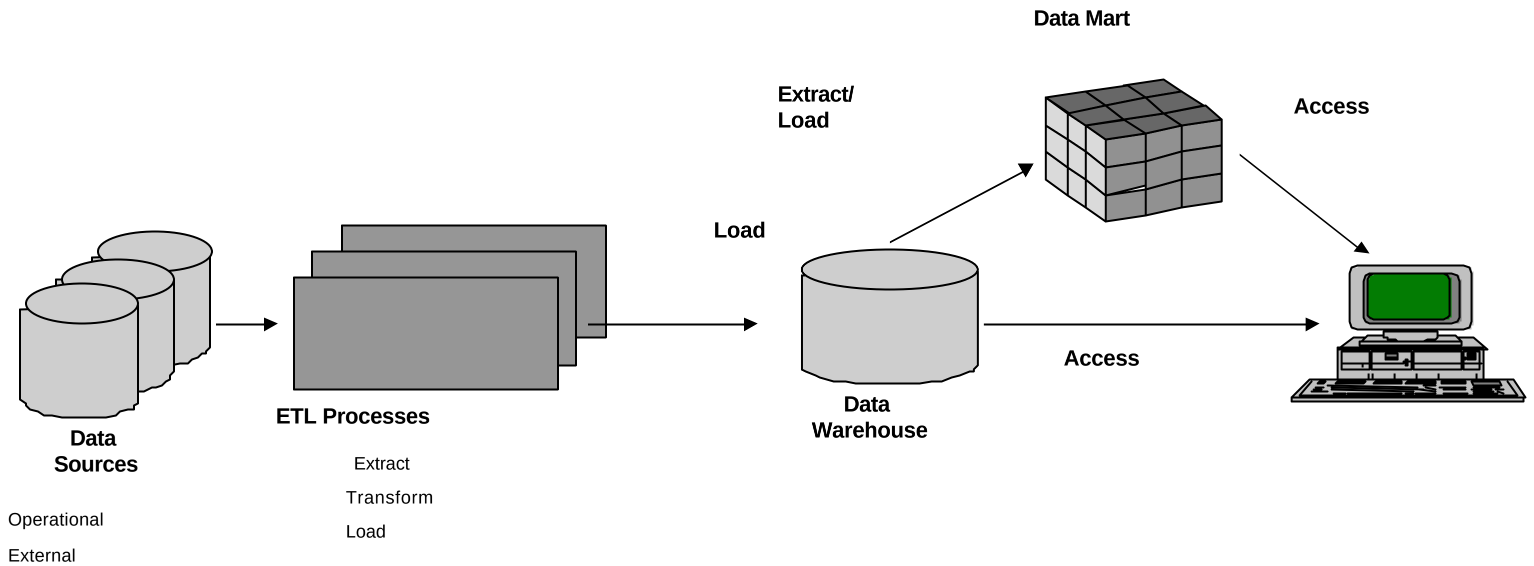
Data Warehousing Architecture Overview (cont.)

Department of Education

Architectural Components:

- *Data Warehouse:* A centralized database that collects, organizes and stores data from operational systems to provide a single source of integrated and historical data for the purposes of end-user reporting and analysis.

A Simplified Data Warehouse Architecture:



Data Warehousing Architecture Overview (cont.)

Department
of
Education

Data Warehouse Terminology:

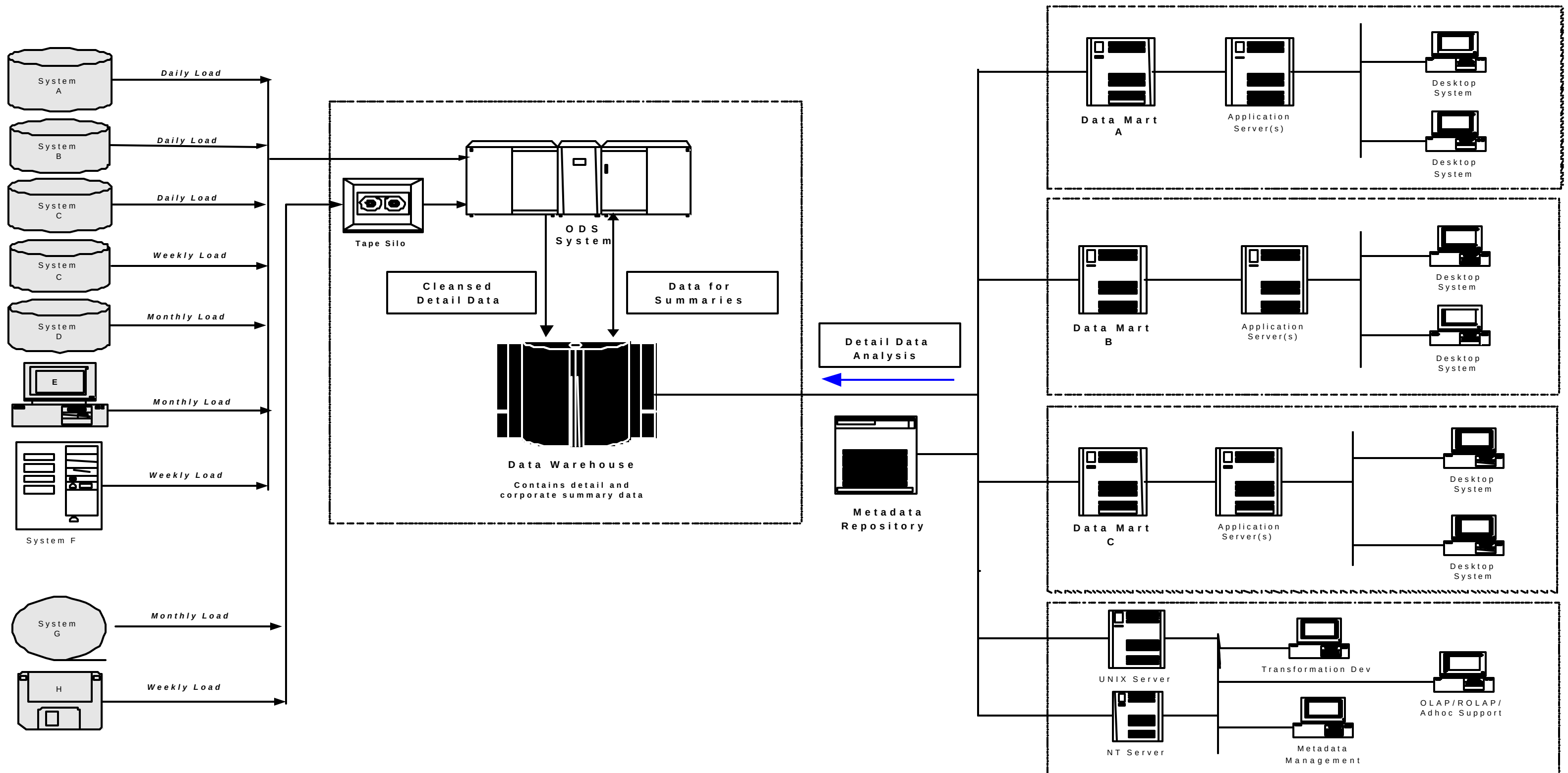
As the data warehouse industry is still evolving, the term "data warehouse" may mean different things to different organizations. The most commonly used meanings within the industry for differing contexts of the term data warehouse are described below:

- *Data warehouse architecture:* Used to describe the end-to-end processes and infrastructure required to move and support data between source systems though the data warehouse to the en-users.
- *Data warehouse:* This term should be used on its own only when describing the centralized database at the core of the data warehouse architecture. The data warehouse becomes a distinct term within environments that contain an Operational Data Store or a Data Mart.
- *Department data warehouse:* Used to describe a warehouse which supports departmental analysis in one to five subject areas, drawing on from one to a few data sources. This definition often overlaps on the low end with the functionality of a data mart. Typical data sizes can range between 10-200GB, usually supporting between 5-50 users.
- *Enterprise-wide data warehouse:* Used to describe a warehouse which provides access to multiple subject areas and users in multiple departments and physical locations. The warehouse is typically fed from one to many data sources to provide a single collection point of information to access and share across the enterprise. There is a wide range of data sizes and number of users that fall within this definition. Data sizes usually range between 50GB - 2TB, supporting between 20 - 2,000 users.

Data Warehousing Architecture Overview (cont.)

Department of Education

Shown below is a sample architecture of a data warehouse environment. Note that the following architecture is described only as an example and is not necessarily a recommended approach in all instances:



SAMPLE ARCHITECTURE

Data Warehousing Architecture Overview (cont.)

Department of Education

The sample architecture has the following features:

- Data from numerous internal operational sources with various formats are loaded at different time intervals
- Data from external sources may also be loaded at certain time intervals (not shown)
- An ODS provides a staging area for the integration of the data warehouse data
- The ODS system also provides the ETL processes, thus eliminating a load from the operational and data warehouse environments
- The Metadata repository is accessible by all users of the data warehouse, thereby defining the meaning of the data warehouse contents for all users
- The application servers can be augmented by web servers (not shown) to provide internet/intranet access to the warehouse for end-users
- The data warehouse contains the current detail data and summaries used at the corporate level
- The data marts contain the departmental level summaries; they may also contain detail data which is only of interest to a particular department. Note that this detailed is not duplicated at the data warehouse level, but it passes through for integrity purposes.
- The centrally located data warehouse is common to many warehouse implementations. The need for physically distinct ODS constructs and multiple distributed data marts vary considerably between projects. Generally, organizational concerns, budgetary constraints, and technological complexity of managing a distributed warehouse greatly favor this approach.

Data Warehousing Architecture Overview (cont.)

Department of Education

Data Distribution and Placement:

The sheer volume of data in the data warehouse constantly pushes the envelope of scalability of the underlying DBMS and hardware. Often, a data warehouse is already on the most powerful hardware and operating system platform that the organization supports, and therefore getting a “bigger box” is not a feasible solution. Technical limitations in today’s hardware and software such as scalability and performance dictate that data placement and distribution be considered a very high priority when making architecture decisions.

Data Modeling for Data Warehousing:

Data modeling is the most fundamental designing task in data warehousing. Data model detail can range from a basic one-page high-level diagram to a detailed model that covers the walls of the data administration office.

Data modeling is nothing new. It has been used in one form or another throughout the history of systems design. However, for data warehousing, data modeling takes on a whole new level of importance. New concepts, terms and techniques now exist to handle the specialized needs of a data warehouse.

Data Warehousing Architecture Overview (cont.)

Department of Education

The data model should function as the key building block to represent the information that supports the business. The information is broken into basic data elements to describe its name, type , relationship, and organization.

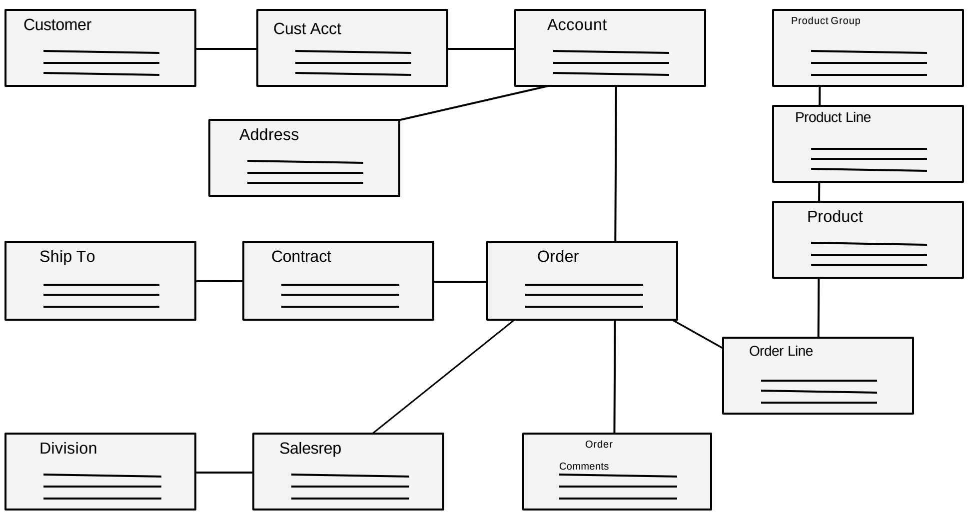
During a data warehouse project, several data models will be created depending on the requirements of the project. As the project complexity increases, a well designed data model becomes the blueprint of the data warehouse requirements.

Specialized modeling tools are used to generate the model diagrams, resulting schema, and sometimes SQL to generate and alter the actual database. However,basic drawing programs can also be used when the scope of the model does not demand robust tools.

Data Warehousing Architecture Overview (cont.)

Department of Education

There are many data modeling disciplines and techniques, chosen generally based on project requirements, the modeler's particular experience and the client's predefined modeling standards. The technique will dictate the final diagram representation. The following data model diagrams are typically seen at data warehousing engagements:



Data Warehousing Architecture Overview (cont.)

Department of Education

Physical Design Issues:

Indexing Strategy: Will both minimize the number of indexes and optimize performance for all critical data accesses. It is tempting to add an index to satisfy every data access that hits a table. However, more indexes per table translate into additional time loading tables and restructuring indexes when inserting and deleting rows. More indexes also mean higher administrative overhead. For example, each index must be considered when sizing and planning for growth.

Partitioning Strategy: Database partitioning is defined as the splitting of tables and indices within a single database; for example the portions of the last past year's data can be partitioned into smaller monthly sales data for faster access.

Population Architecture

Department of Education

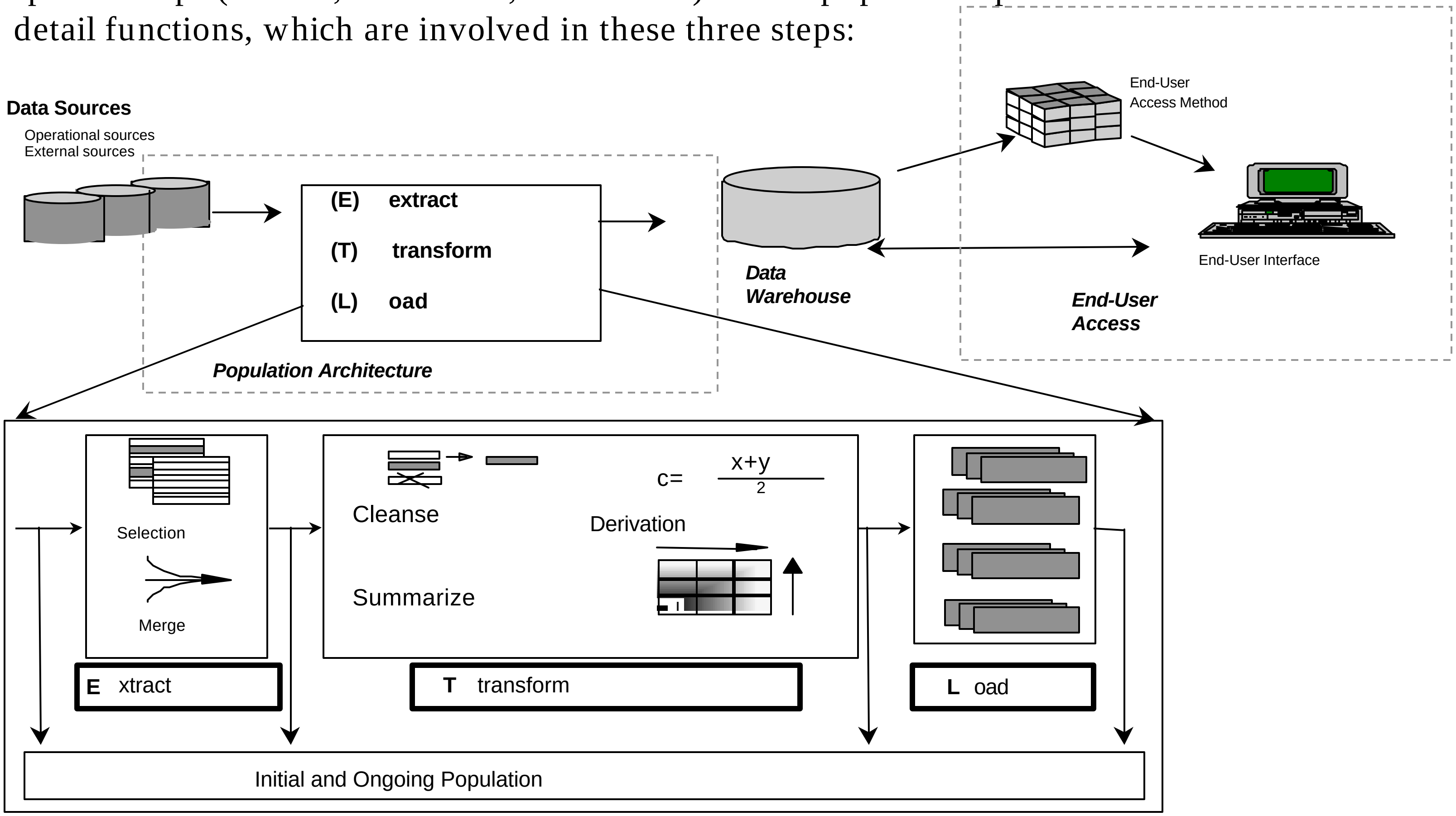
- This section discusses the process of collecting and moving data from various sources into the data warehouse. It describes the steps and techniques for this process and attempts to detail many of the issues and problems. It clarifies many of the frequently asked questions about the population process and increases the reader's understanding of population architecture requirements.

- According to industry experience, nearly 80% of the effort required in a data warehouse engagement revolves around the population process. This is due to several factors:
  - Identification of relevant source data
  - Assessment of the data quality and cleansing of data
  - Data mapping from source to data warehouse
  - Design and creation of the data model and database
  - Technical effort required in physically extracting the operational data and deliver it into the data warehouse

Population Architecture (cont.)

Department of Education

The following depiction of the conceptual architecture of a data warehouse system illustrates the specific steps (Extract, Transform, Load - ETL) for the population process. It also demonstrates the detail functions, which are involved in these three steps:



Population Architecture (cont.)

Department of Education

Architectural Options:

Process Distribution:

Transform in the data warehouse environment - The entire Transform process occurs in the data warehouse. The extraction is accomplished in the data source environment and the load is performed in the central data warehouse environment.

Transform in the data source environment - The Transform process occurs on the data sources. The extraction is accomplished in each source environment and the extracted data is sent to the Transform step within the same environment.

Transform in a separate environment - The Transform process occurs in a separate environment. The extraction is accomplished in each source environment and the extracted data is sent to the Transform step within the transform server. This server transforms the extracted data.

Population Architecture (cont.)

Department
of
Education

Update Techniques:

There are several timing approaches to assist in transferring data between two environments, because some data in the data warehouse needs to be updated immediately after a change in the source systems, whereas other approaches update changes on a daily or weekly basis:

Real-Time - Provides real-time updates to the data warehouse. An example of real-time data capture includes sending updates to the data warehouse as soon as the data is processed in the OLTP environment. Common real-time approaches include the use of database gateways. This update approach is often used when the data warehouse provides direct feedback to operational systems.

Batch-Scheduled - Provides data delivery at a predefined time period. Common batch-scheduled approaches use gateways or data loader routines. An example of batch-scheduled data capture includes sending sales balance data updates to the data warehouse at the end of the day, week, or month.

Batch-Event - Provides data deliver after a specific event. Common batch-event approaches use gateways. An example of batch-event data capture includes sending sales data immediately after a sales transaction has been registered.

Population Architecture (cont.)

Department of Education

Initial and Ongoing Population:

Once the data warehouse architecture is designed, attention shifts to the population of the data warehouse. There are two principal strategies for accomplishing this:

Full population - Extracting the entire set of data stored in the data sources and transferring it to the Transformation step, thus starting the data warehouse with the current step of OLTP data

Incremental population - Incrementally updating the data warehouse while relevant transactions occur on the operational system.

Extraction Techniques:

After initial population strategies are reviewed, several common techniques to extract data from the data sources can be considered. This section describes and lists decision criteria. It also discusses the replication of data sources.

Population Architecture (cont.)

Department of Education

Full Extract - Data is entirely extracted from its sources, then, the Transform or Load step determines which data is needed. This is required for full population of the data warehouse.

Delta Extract - Only data which has been changed, inserted or deleted since the last extract is taken from the data sources. This typically isolates the needed data for the Transform and Load steps. The Delta Extract techniques can be classified as system level, table level and row or record level.

ETL Tools:

Traditionally, Extract and Transform solutions have been custom-coded for each project. These custom solutions can be inefficient to develop and result in a stand alone extraction program for each data source, making them difficult to manage. In an effort to reduce the development time and improve manageability, software vendors developed packaged Extract/Transform/Load (ETL) applications to help control this major data warehouse expenditure

Population Architecture (cont.)

Department of Education

Requirements for ETL Tools

Data transformation is a mainstay of many common IT projects, from application downsizing and package replacements to Year 2000 transformations to data warehousing. While data transformation is not inherently difficult, it can be expensive in a large enterprise due to several factors:

- Large numbers of data sources can result in numerous transformation modules
- Frequent addition of new data sources and modifications to existing sources can result in high maintenance costs
- Stringent performance, scalability, and robustness requirements are difficult to meet in an on-going fashion where complex work flows combine with complex configuration control requirements to challenge the management of the ETL process

End-User Access

Department of Education

Through presentation and analysis tools, end-user applications provide users access to information contained within a data warehouse or data mart to help them leverage people and information for increased business performance. These applications fall into many different categories, as distinguished by their various capabilities. However, their specific uses depend upon the requirements and knowledge levels of their users. Because they may vary greatly, it is uncommon for one end-user application to satisfy all requirements. As a result, an organization will commonly use more than one application to fulfill end-users' requirements.

There are several categories of tools designed to access and analyze data which present advantages and disadvantages to users, depending upon their needs. These categories are:

Reporting tools - Aimed at less technical users usually at both ends of an organization, reporting tools are designed to facilitate the production and distribution of reports containing generally static warehouse information in the form of graphs, charts, and tables.

End-User Access - (cont.)

Department of Education

Query tools - Querying capabilities are used to support the retrieval of basic information through Structured Query Language (SQL) or a dynamic data exchange (DDE). Users proficient with query languages and database structures may write their own retrieval code transparently using familiar business terms easily selected from menus or objects. These tools block out the complex technical structure and present information in simple and effective business formats.

Analysis tools - Allow users to view historical data at various levels of detail (or granularity) to examine trends, establish rankings, and / or examine the effects of specific business decisions.

Knowledge Discovery tools - Knowledge discovery tools differ from query and analysis tools in that they, rather than users, ascertain important trends and patterns from information stores. These tools can be highly useful, as human beings are limited not only by their inability to differentiate between mere correlations and deeper cause and effect relationships, but also by their inability to determine the relevancy of patterns derived from historical data and extrapolate them into the future. They include data mining tools, impact analysis tools, risk analysis tools, and time series analysis tools.

End-User Access - (cont.)

Department of Education

Data Mining:

Data mining differs from traditional decision support systems in its attempt to interpret data with pattern recognition technology. Discovering relevant patterns in large data stores (at atomic levels) can be important for more effective decision making. More advanced data mining detects patterns in the data that are unknown to the end-user, generating hypotheses rather than reporting or confirming them. Primitive data mining searches for patterns requested by the end-user. Its Return on Investment (ROI) emerges from a reduced cost of analysis and, most importantly, through the discovery of important cost-saving or revenue-enhancing trends.

Examples of data mining techniques include:

- Neural networks
- Case-based reasoning
- Rule induction
- Genetic algorithms
- Statistics (multivariate regression)
- Compression and regression trees

End-User Access - (cont.)

Department
of
Education

End-User Access Tools Architectures:

System architecture plays an important part in the selection and evaluation of end-user tools, affecting performance, remote-use, and stability.

Client/Server architectures - Divides the application architecture into two or three tiers. With the data warehouse repository on one machine, designers can choose between end-user access applications residing entirely on client machines or splitting them into client and server portions.

Net-centric architectures - Allows access to the data warehouse through the World Wide Web. Companies involved in data warehousing are increasingly turning to net-centric architectures, generally web-enablement, to magnify their competitive advantage. The tools used to accomplish this are new; therefore, the products and architectures supporting them are neither mature nor robust. Even as vendors continue to develop and expand this market, their clients are beginning to realize significant benefits.

Security issues - Makes sure repository information is kept private and secure.

Executive Summary Overview

Department of Education

- *Integration Application and Technical Architecture Standards*
  - Introduction
  - Standards for Legacy Systems and Network Analysis Phase
  - Standards for Configuration Phase
  - Standards for the Build and Test Phase

- *Data Warehouse Application and Technical Architecture Standards*
  - Introduction
  - Data Movement Standards
  - Data Warehousing Architecture Overview
  - Population Architecture
  - End-User Access

- ***Internet Application and Technical Architecture Standards***
  - **Introduction**
  - **Standards Covering Aspects of Development**
  - **Construction Standards**

Introduction:

SFA has decided to implement a standardized software development and deployment architecture for all new software development going forward. This is in contrast to the past practice of "anything goes," where software developers were allowed to use nearly any combination of hardware, operating system, communications protocols, and programming language they wished. While convenient for the software developers, this practice has saddled SFA with a complex, heterogeneous environment that is expensive to maintain and difficult to modify as business needs evolve. To break from this practice SFA has decided to adopt a net-centric, multi-tiered, component-based application environment.

All new custom-developed software - and to the degree possible, all commercial off-the-shelf software - will utilize common internet standards like TCP/IP, HTTP, SLL, HTML, and ECMAscript for communications and presentation. For applications that do not require a user interface, like business-to-business or eCommerce data exchanges, XML will be used. Other standards like LDAP, X509, and ANSI SQL '92 will also be used where applicable to provide services throughout the enterprise.

Internet Application and Technical Architecture Standards (cont.)
Introduction

Department of Education

Application functionality will be distributed across several tiers. User interface presentations will be handled by thin clients like web browsers or lightweight client applications. Static information resources are hosted on a web server. Presentation logic, application-specific workflow, and reusable business logic will be encapsulated into components that are managed by an application server. Additional, special-purpose servers can be relied upon to provide enterprise-wide services for security, authentication, personalization, and business-to-business exchange. Existing enterprise data resources will continue to be housed in a separate legacy layer that ins accessed via the EAI layer. Storage for new data will reside in the Internet layer.

As much reusable business logic as possible will be encapsulated into components, which are independent objects that are dynamically and transparently executable over a network by multiple different applications simultaneously. These components reside on a dedicated server called an application server. The repository of components on the application server will allow new applications to rapidly and efficiently leverage earlier development efforts, facilitating a “plug and play” approach to building applications.

Internet Application and Technical Architecture Standards (cont.)
Standards Covering Aspects of Development

Department of Education

All developers will follow the naming and commenting conventions outlined in the Java Code Conventions documentation maintained by Sun Microsystems.

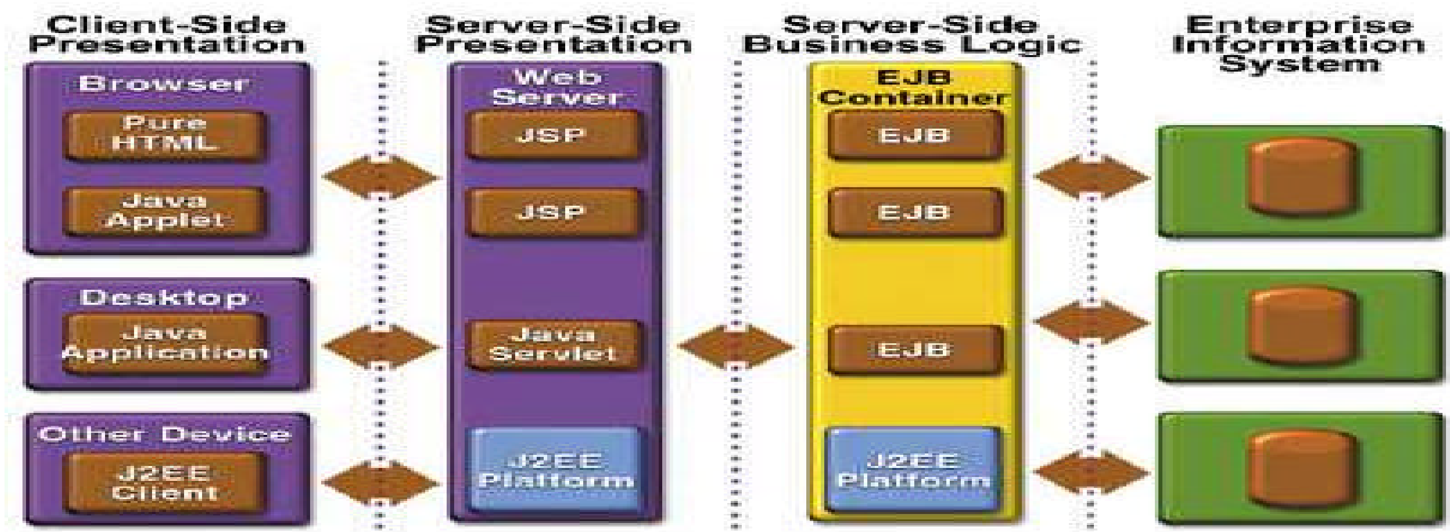
The SFA's standard method for securing TCP/IP based transactions is called Secure Socket Layer (SSL). SSL establishes a framework in which a cryptographically-secure network connection between the browser and the web server can be made. This allows the browser and server to exchange data without allowing others on the network to see the cleartext of the data transmitted, and it also ensures the reliability of the data. SFA must also utilize products that are compliant with the FIPS-140-1 standard for cryptographic security.

If the data being transferred between the application and the user is sensitive in nature, the data transfer will be cryptographically secured with SSL v3. For data connections between components of the application, or between the application and enterprise-wide services like an LDAP server, SSL encryption will be used if the data is considered sensitive or if the transaction can potentially provide access to data considered sensitive.

Internet Application and Technical Architecture Standards (cont.)
Browser-Based Application Construction Standards

Department of Education

The standard architecture for the development of all applications is based on the Model, View, Controller (MVC) design pattern. Models are the enterprise data and business logic resources, and are typically implemented as EJBs. Views are static HTML templates that include JSP code to provide users with a dynamic view of the data and business logic contained in the EJB models. Controllers are servlets that mediate between the models and the views, providing request handling and workflow management within the application.



The Model components are in red, the View components are in blue, and the Controller components are in green.

Java Applet Construction Standards:

Much of the philosophy behind the preceding standards discussion regarding client-side presentation is applicable to the inclusion of Java applets in Web pages. In particular, it is necessary to segment the potential user population based on their browser's capabilities and to then target the applet's development to those technology constraints.

- For sites that are accessible to the public, applets are forbidden. The delays experienced when downloading an applet and initializing the browser's Java Virtual Machine (JVM) are frustrating to the user.
- For sites that are restricted to business partners, applets may be used if necessary, but they aren't encouraged. It is strongly suggested that developers use the downloadable Java Plug-In product with the Java Runtime Environment v.1.2 for execution consistency and guaranteed access to Java 2 functionality.
- For sites that are restricted to SFA employees, applets may be used if necessary. It is strongly suggested that developers use the downloadable Java Plug-In product with the Java Runtime Environment v.1.2 for execution consistency.

Java Application Construction Standards:

Stand alone Java applications, if developed, will be based on the Java 2 Standard Edition APIs. Java applications will not take advantage of other APIs unless they are 100% Java, and even in that case, those APIs should not be used if the core or extended APIs are suitable for the task. As with the Web-based applications, Java applications will utilize the MVC design patter architecture.

Internet Application and Technical Architecture Standards (cont.)
Construction Standards

Department
of
Education

Faceless Application Construction Standards:

"Faceless" applications are those applications that are meant to facilitate automated business functionality without any direct user involvement. An example of this kind of application is a servlet that receives an XML document, parses it, updates a database with data from the XML document, and returns a confirmation message to the original sender. Thus they can be considered to implement (and be subject to the architecture standards relevant to) the Model and Controller aspects of the MVC design pattern.